一、数据库性能监控

1. 数据库性能可能出现的问题

数据库本身的性能问题

数据库的运行环境即服务器操作系统的相关性能问题

物理环境问题，例如网络问题等

1. 监控工具

性能监视器

**doDBA tools 一款免费的实时数据库监控工具**

<https://github.com/dblucyne/dodba_tools.git>

其他推荐：

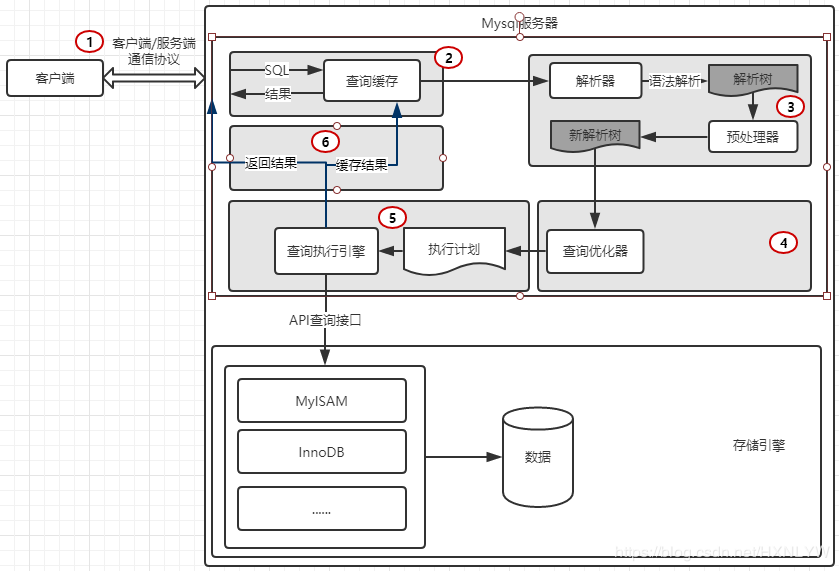
<https://www.toutiao.com/a6654350892373901831/?tt_from=mobile_qq&utm_campaign=client_share&timestamp=1549354401&app=news_article&abtest_id=1&iid=59568063679&utm_medium=toutiao_android&utm_source=mobile_qq&group_id=6654350892373901831>

1. 常见监控指标



二、Sql语句性能调优

1. 执行一条sql查询语句时mysql处理过程



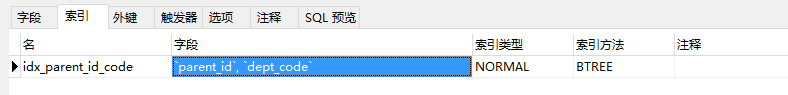
1. Sql查询语句各部分在mysql中的执行顺序



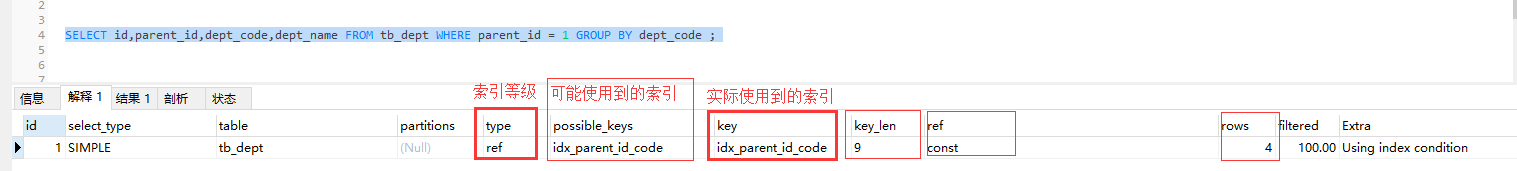
1. 一个例子

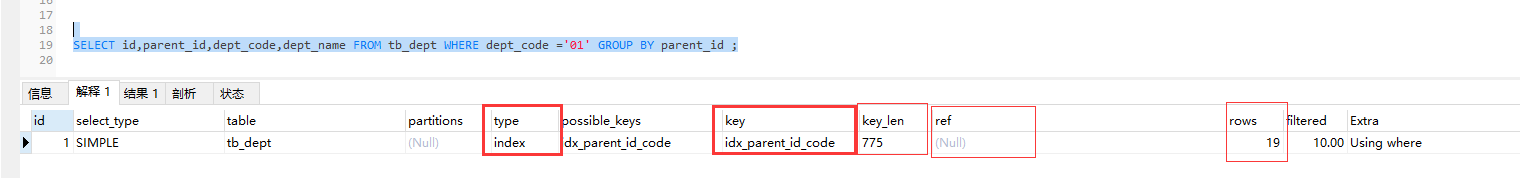


在tb\_dept中建立一个复合索引 idx\_parent\_id\_code：



然后看下两个sql 解释的结果：

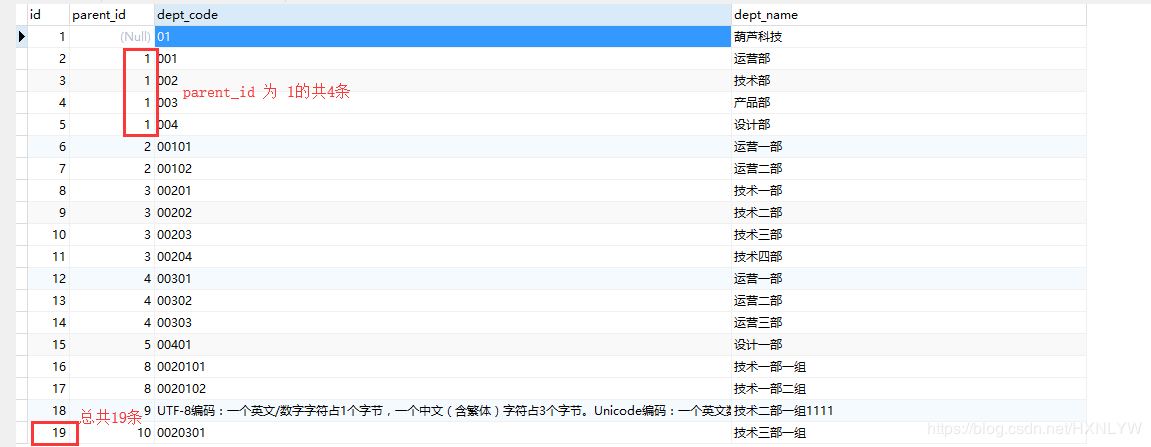




1）在当前索引下，哪一个sql索引利用率高？

借助于上文中查询SQL的执行顺序，是先执行 WHERE再执行 GROUP BY 的，即：第一个sql执行的顺序是先执行了 where后的 parent\_id 然后执行了 group by 后的 dept\_code，顺序是和索引的顺序是一致的，type等级为ref，扫描行数rows为 4；而第二个sql是先执行了 where后的 dept\_code 然后执行了 group by 后的 parent\_id，顺序是和索引的顺序是不一致的，type等级为index，扫描行数rows为 19；从解释结果看，第一条的sql索引利用率高于第二条的。

或者从扫描的行数rows对比数据源也可直观的看出，两个语句的性能：



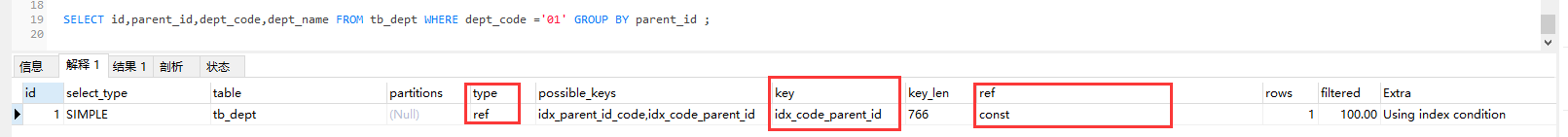
2）怎么优化？

如果业务中用到第二个sql，那么就需要调整索引的顺序和sql执行顺序一致。

或者两个sql都用到了，那么就再建一个复合索引 （idx\_code\_parent\_id）



然后再看下第二条的执行计划：



1. EXPLAIN用法

查看优化器如何决定执行查询的主要方法，可以帮助我们深入了解MySQL的基于开销的优化器，还可以获得很多可能被优化器考虑到的访问策略的细节，以及当运行SQL语句时哪种策略预计会被优化器采用。会返回执行计划的信息，而不是执行这条SQL（如果 from 中包含子查询，仍会执行该子查询，将结果放入临时表中）。

EXPLAIN输出字段包含id、select\_type、table、type、possible\_keys、key、key\_len、ref、rows、Extra几项信息。

各字段含义

Id：select查询的序列号，包含一组数字，表示查询中执行select子句或操作表的顺序

select\_type：表示对应行的查询类型是简单查询还是复杂的查询

SIMPLE 简单的select查询，查询中不包含子查询或者UNION

PRIMARY 查询中若包含任何复杂的子部分，最外层查询则被标记为PRIMARY

SUBQUERY 在SELECT或WHERE列表中包含了子查询

DERIVED 在FROM列表中包含的子查询被标记为DERIVED（衍生），MySQL会递归执行这些子查询，把结果放在临时表中

UNION 若第二个SELECT出现在UNION之后，则被标记为UNION：若UNION包含在FROM子句的子查询中，外层SELECT将被标记为：DERIVED

UNION RESULT 从UNION表获取结果的SELECT

Table：表示对应行正在访问哪一个表，表名或者别名

TYPE：显示的是访问类型，是较为重要的一个指标，结果值从好到坏依次是：system > const > eq\_ref > ref > fulltext > ref\_or\_null > index\_merge > unique\_subquery > index\_subquery > range > index > ALL ，一般来说，得保证查询至少达到range级别，最好能达到ref；

POSSIBLE\_KEYS：显示查询有可能会使用到哪些索引，表示该索引可以进行高效地查找，但是列出来的索引对于后续优化过程可能是没有用的，只是一种预测

KEY：显示MySQL实际决定使用的键（索引）

KEY\_LEN：key\_len列显示MySQL决定使用的键长度。如果键是NULL，则长度为NULL。使用的索引的长度。在不损失精确性的情况下，长度越短越好。

REF：显示使用哪个列或常数与key一起从表中选择行

ROWS：显示MySQL认为它执行查询时必须检查的行数。注意这是一个预估值。

EXTRA：显示MySQL在查询过程中的一些详细信息，MySQL查询优化器执行查询的过程中对查询计划的重要补充信息

1. sql语句性能优化策略

<https://www.cnblogs.com/SimpleWu/p/9929043.html>

<https://zhuanlan.zhihu.com/p/86466788>

三、慢查询日志或执行计划（oracle）

1. 简述：MySQL的慢查询日志是MySQL提供的一种日志记录，它用来记录在MySQL中响应时间超过阀值的语句，具体指运行时间超过long\_query\_time值的SQL，则会被记录到慢查询日志中。long\_query\_time的默认值为10，意思是运行10S以上的语句。默认情况下，Mysql数据库并不启动慢查询日志，需要我们手动来设置这个参数，当然，如果不是调优需要的话，一般不建议启动该参数，因为开启慢查询日志会或多或少带来一定的性能影响。慢查询日志支持将日志记录写入文件，也支持将日志记录写入数据库表。
2. 一些参数及其含义

slow\_query\_log ：是否开启慢查询日志，1表示开启，0表示关闭。

log-slow-queries  ：旧版（5.6以下版本）MySQL数据库慢查询日志存储路径。可以不设置该参数，系统则会默认给一个缺省的文件host\_name-slow.log

slow-query-log-file：新版（5.6及以上版本）MySQL数据库慢查询日志存储路径。可以不设置该参数，系统则会默认给一个缺省的文件host\_name-slow.log

long\_query\_time ：慢查询阈值，当查询时间多于设定的阈值时，记录日志。

log\_queries\_not\_using\_indexes：未使用索引的查询也被记录到慢查询日志中（可选项）。

log\_output：日志存储方式。log\_output='FILE'表示将日志存入文件，默认值是'FILE'。log\_output='TABLE'表示将日志存入数据库，这样日志信息就会被写入到mysql.slow\_log表中。MySQL数据库支持同时两种日志存储方式，配置的时候以逗号隔开即可，如：log\_output='FILE,TABLE'。日志记录到系统的专用日志表中，要比记录到文件耗费更多的系统资源，因此对于需要启用慢查询日志，又需要能够获得更高的系统性能，那么建议优先记录到文件。

1. 开启慢查询日志

>set global slow\_query\_log=1(或on);

设置时间临界点，下述语句表示：执行时常超过1s的语句都将被记录到慢查询日志文件中

>set long\_query\_time = 1;

设置慢查询存储的方式

>set globle log\_output = file;

查询慢查询日志的开启状态和慢查询日志储存的位置

>show variables like '%quer%';

参数说明：

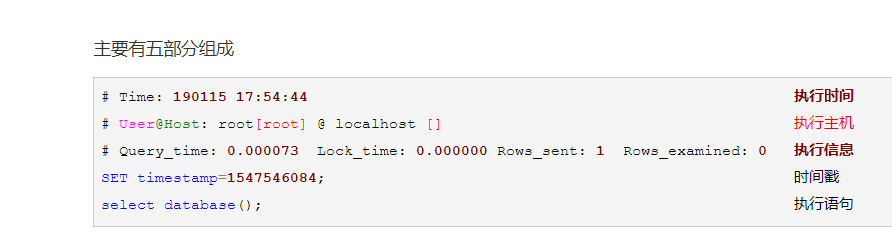
slow\_query\_log : 是否已经开启慢查询

slow\_query\_log\_file : 慢查询日志文件路径

long\_query\_time :  超过多少秒的查询就写入日志

log\_queries\_not\_using\_indexes 如果值设置为ON，则会记录所有没有利用索引的查询(性能优化时开启此项,平时不要开启)

1. 存储格式



1. 学习链接，更多解析日志工具

<https://www.cnblogs.com/splendid/p/10272951.html>

<https://www.cnblogs.com/luyucheng/p/6265873.html>